First Hit Fwd Refs

Generate Collection Print

L5: Entry 2 of 4 File: USPT Dec 30, 2003

DOCUMENT-IDENTIFIER: US 6671818 B1
TITLE: Problem isolation through translating and filtering events into a standard object format in a network based supply chain

Detailed Description Text (1250):
On the Internet, an agent 9300 (also called an intelligent agent) is a program that gathers information or performs some other service without the immediate of a user. See FIGS. 90 and 93. Typically, an agent, using parameters provided by the user, searches all or some part of the Internet, gathers the requested information, and presents it back to the requesting user. Intelligent external agent technology will continue to grow as the eCommerce market develops. As the market saturates with products and information, the need for techniques or agents to filter this information will grow.

Detailed Description Text (1322):
This technique filters the information or content displayed to the user based on what is entered by, or known about the user. Many personalized sites use some form of content filtering. A variety of different filtering techniques have emerged. Some are adaptations of traditional client/server techniques adapted to the Web. In the first grouping of techniques presented here, the user controls how the information is filtered. User-controlled, explicit content filtering. Techniques in this category enable the user to filter data or content.

Detailed Description Text (1338):
Push technologies enable an enterprise to reach and provide value to their customer outside of the traditional interactions. Of the personalization techniques described above, content filtering doesn't require that it be delivered though an interactive session. Keep in mind that push does not necessarily mean personalized. A site may simple push the same information to all users or subscribers. Personalized push refers to information that is filtered based on a specific user's request or profile or where the delivery is scheduled for a specific user. There are a variety of different legitimate business reasons to push a users information. The following are some examples of personalized push: Information Delivery. A user subscribes to receive information on a scheduled basis. The information that is pushed is either determined through user selection--the user selects the subjects and type of information that they wish to receive, or the site determines what information the user may be interested in receiving based on their profile information. For example, at an investment web site, users are allowed to subscribe to investment information feeds. The user decided to receive a daily feed of the stock quotes for the commodities they hold. The site pushes this information and additionally pushes news articles and stock recommendations based on the user's portfolio holdings. Event Reminders. The user subscribes to receive event reminders from the web site. An event reminder might remind the user of specific dates of interest such as a birthday or anniversary, a bill reminder or that an action is required in the near future. Information Update. A site may also push updates to a user. For example, Microsoft's Expedia allows users: to request fare updates. A user can choose a specific destination of interest. If the fare to this location changes, the user is notified of the fare update. Don't be a junk e-mailer. The push medium is powerful and potentially less costly than conventional mail. This doesn't justify its use as junk mail. It does not work because everyone receives

junk mail, and junk mail that ends up in the trash is failed target marketing efforts. Personalize it. Deliver valuable information that the customer is interested in. Allow them to select the topics, how to filter the content and the frequency.

Detailed Description Text (1459):
Payment 10604 After a total has been established, a payment method must be determined. A variety of mediums can handle the transfer of money. The methods, flow, technology, and potentially messaging, will vary by implementation. Issues concerning security, liability, and relationship to fulfillment need to be worked out. Listed below are some considerations for determining the payment flow and mediums to be utilized. Anonymity. If there is a need to allow the users to remain anonymous, an anonymous medium may need to be implemented. Implementations such as a silent bidding site may require strict standards and mediums for anonymity. In general, anonymity is not a concern for most implementations. Monetary Transaction Size. If the site will be handling very small or very large monetary transactions additional considerations will be required. Sites accepting micro value transactions will need to plan a process to collect and verify the payment. To make low-value transactions cost effective, solutions may sacrifice security. In some implementations, it is assumed "some" fraud will occur but in such small denominations as to be negligible. Transaction Cost. Depending on the payment method, there are numerous potential associated costs. Most mediums have either transaction costs or may involve a broker requiring additional fees. Understanding the costs associated is important when planning an efficient payment system. Audit Trail. Some implementations may record each transaction with a unique identifier used to track funds if necessary. Security. In the past, eCommerce has been hampered by the absence of secure and robust transaction options. Recent development of secure online payment options over the Internet have been a primary enabler. Strongly-encrypted online purchase transaction protocols have been developed and integrated into software for consumers, merchants, and banks to enable secure credit card transactions. Consumer Type. The types of flow and payment medium will vary greatly depending on the consumer or purchaser. B-C implementations require payment (or at least authorization) once the order is placed. For the buyer-centric, trading partner relationship, the established infrastructure may handle payments using traditional invoicing or an Internet-enabled form of EDI or EFT (Electronic Funds Transfer). Electronic Authentication. Some sort of digital signature strategy would need to be in place between trading partners and potentially the financial institution. Message Standards. Payment instructions must be recognizable to all parties involved. Payment Methods There are a multitude of different vendors and technologies available for handling electronic payments. The infrastructure, process, and technology may vary dramatically from vendor to vendor. The actual mediums for the current payment options fall into these categories: Credit-Based Payment. Today, the most widely-used electronic payment option is the credit card. With the new transaction protocols and security features, credit cards can be used on the Internet just as they are in the real world. Consumer confidence is higher with the already familiar standard. Current overhead for clearing, settlement and fraud makes credit card based solutions uneconomical for transactions of small dollar amounts. Debit-Based Payment. Payment utilizing this method will directly debit and credit accounts. These may take the form of debit cards, electronic checks or messages utilizing EDI or EFT. Electronic Cash. Electronic cash is the electronic equivalent of real paper cash. It is usually implemented using public-key cryptography, digital signatures and blind signatures. Electronic cash is "digital" money on the computer's hard disk. Theoretically, the money could be spent in very small increments, such as tenths of a cent (U.S.) or less. In an electronic cash system there is usually a bank, responsible for issuing currency, consumers that obtain cash from either banks or brokers and merchants who will accept the digital cash for goods and services. In short, the bank, merchant and consumer each own a public and private key which is used to encrypt and digitally sign the electronic cash. Smartcards. A smartcard is a programmable storage device the same in size and appearance as a

235 t

normal credit card. It contains a microchip to store and process information. Some of these cards can contain stored value in the form of digital coins. A lost card means lost value, just like cash. The person holding the card can spend the value stored on it at any merchant accepting smartcards. This technology is particularly useful for online shopping, and is far less vulnerable than systems storing value on a hard disk. Transaction costs for this form of payment are very low, enabling the user to conduct micro-transactions of one penny or less. Microsoft and several computer manufacturers are pushing for standards to incorporate smartcard readers into PC keyboards, and most TV set-top Internet access devices already have them Digital Wallets. Digital wallet software facilitates secure, online transactions between the consumer and the merchant, and between the merchant and the bank. For the consumer, there will soon be literally hundreds of software "digital wallets" available. They will likely be free and similar in function, running within a web browser. Payment Authorization. In many cases, consumer sites which implement a credit! card payment method will require payment authorization. In some cases the actual settlement process can not occur until the items are shipped. JEPI, being developed by W3C and CommerceNet in cooperation with many large technology companies, is a standard mechanism for Web clients and servers to find out what payment capabilities they have in common, and negotiate the payment instrument, protocol, and transport between one another. This will be transparent to the user; they will simply be told by the wallet software what payment options are available at this merchant (along with any available discounts for payment type or membership affiliations), and asked to choose.

L19: Entry 8 of 13 File: USPT Jun 4, 2002

DOCUMENT-IDENTIFIER: US 6398105 B2
TITLE: Automatic data collection device that intelligently switches data based on data type

Brief Summary Text (5):
An ADC device platform having bar code reader adeptly accesses and retrieves data stored in the form of a bar code label. Data representing virtually any product or service found in the stream of commerce may be encoded in a bar code label for later access by an ADC device platform having a bar code reader. Bar code readers include laser scanners as well as other means of collecting product information, such as a bar code wand, a still camera or an area imager. In addition to bar code labels, other ADC data formats include Radio Frequency ("RF") tags, resonators, SmartCards, magnetic strips, Optical Character Recognition ("OCR"), speech input two-dimensional ("2D") symbols, dipole devices (such as those recited in U.S. Pat. No. 5,581,257), and any symbol having encoded data therein.

Brief Summary Text (13):
ADC devices accommodated by embodiments of the invention include bar code readers, speech recognition systems, RF tag readers, resonator readers, SmartCard readers, two-dimensional symbol readers, ASCII data devices, AIMI-ECI data devices, dipole device readers, and optical character recognition ("OCR") systems. Data may be communicated to remote and local applications using any data protocol, including the Transmission Control Protocol ("TCP"), the User Datagram/Internet Protocol ("UDP/IP), or the User Datagram Plus Protocol ("UDP+"), which is noted below.

Detailed Description Text (7):
An exemplary ADC device platform, such as the ADC device platform 100, includes communications device 116, a computing system 120, and the ADC devices 117, 118. The ADC device platform 100 may comprise more than two ADC devices as indicated by the ellipsis between the ADC device 117 and the ADC device 118. The ADC devices 117, 118 may comprise, for example, bar code readers, radio frequency ("RF") tag readers, resonator readers, SmartCard readers, magnetic stripe readers, two-dimensional ("2D") symbol readers, optical character recognition ("OCR") readers, ASCII data devices, AIMI-ECI data devices, speech input recognizing devices, text-to-speech recognition devices, dipole device readers, an all other forms of scanning or imaging devices. AIMI-ECI ("extended channel interpretation") utilizes symbol value ranges (e.g., the range 00000 to 811,799) that represent particular classes of items (e.g., retail merchandise). An exemplary RF tag reader suitable for use in the ADC device platform 100 is described in U.S. application Ser. No. 08/771,320, entitled, "Automatic Mode Detection and Conversion System for Printers and Tag Interrogators," filed on Apr. 27, 1998 and assigned to a common assignee. The ADC device 117 may be a different type of, or the same as, the ADC device 118.

Detailed Description Text (11):
If the computing system uses a non-Windows operating system, then a TCP/IP sockets interface will be used. Sockets provide an identifier for a particular service on a particular node of a network. The socket consists of a node address and a port number that identifies the service. The Transmission Control Protocol ("TCP"), governs the break up of data messages into packets to be sent via the Internet Protocol ("IP") and the reassembly and verification of the complete messages from

packets received. The ADC data server 130 allows multiple clients, such as the remote application 109 and the local application 111, to access multiple ADC devices without burdening these client applications with an understanding of the low level ADC device protocols or how to share access to multiple ADC devices.

Detailed Description Text (14):
The ADC data server 130 communicates with remote ADC clients, such as the remote application 109, through the network communications unit 221. In a preferred embodiment, remote ADC clients communicate with the ADC data server 130 using the Winsock 1.1 socket's interface over TCP/IP. Winsock is an application programming interface ("API") that provides a TCP/IP socket interface in the Windows operating system. Embodiments of the network communications unit 221 may utilize a variety of communications methods in communicating with remote applications, including sockets, TCP/IP, UDP, and UDP+.

Detailed Description Text (20):
The ADC protocol handlers in the ADC protocol handler collection 232 provide APIs that allow applications to retrieve ADC data and control ADC devices, such as the ADC devices 117, 118. Each ADC protocol handler in the ADC protocol handler collection 232 is a COM object that supports an ADC device-specific interface to guarantee access to the interface for the ADC data server 130, according to an embodiment of the invention. The ADC device-specific interface operations include opening a client communications channel to a specific ADC device, such as the ADC device 117. For ADC devices that support client handles, the ADC device-specific interface issues a device request that causes the ADC device to return a client handle. A client handle is a numeric value used by the device to identify specific clients. A client handle is assigned to an application when the application first requests a communications channel with the device. A client is a single instance of an application that communicates with the device. For ADC devices that do not support handles, the corresponding ADC device handler itself may generate a handle, according to some embodiments of the invention.

Detailed Description Text (24):
The ADC data server 130 intelligently routes data to one or more clients based upon routing data stored in an ADC data grid, shown and described below with respect to FIG. 3. The ADC data grid operates as a data filter. The ADC data server 130 supports data filtering so that the data sent to ADC clients matches their requested grid criteria. The ADC data server 130 supports non-ADC device-specific grid criteria, such as a general request for a particular data type from any ADC device. The ADC protocol handlers support ADC device-specific grid criteria, such as a request for a particular data type for a particular ADC device. The ADC data grid may be changed dynamically. The ADC data grid may comprise three components, according to an embodiment of the invention. The three components are one or more data classes, a device-independent grid data mask, and a device-dependent grid mask. A device-dependent mask is specific to the device type. For example, a device-dependent mask may exist for Code 39, which is a type of data that may be produced from an ADC device known as a bar code scanner. Non-device specific grid criteria may be represented in a data mask, having a data pattern such as "###-##-####" where "#" represents a numeric value and "-" represents a dash literal. A literal is a value used in a program that is expressed as itself rather than as a variable's value or the result of an expression. The ADC data server 130 finds a match when an ADC data grid specification matches received data. To perform grid matching, the ADC data server 130 first attempts to match one of the data classes and then tries to match the device-dependent grid and the device-independent grid, according to an embodiment of the invention. If all three conditions are satisfied, then the data response will be returned to the client. In other words, the grid specifications may be logically "ANDed" to determine whether the ADC data grid matches the received data.

Detailed Description Text (28):

The ADC data server 130 uses shared memory and process synchronization objects to perform inter-process communication ("IPC"), such as communications with an ADC device driver for an ADC device. The IPC mechanisms are hidden within the ADC data server's API, and the ADC device handlers. The ADC data server API provides an ADC device-ADC data server interface. The ADC device-ADC data server interface hides the IPC mechanism from the ADC data server process. The ADC device-ADC data server interface initializes and deletes an ADC data server API COM object for each ADC device and opens and closes a logical communications channel with each ADC device, such as the ADC device 117. The ADC data server 130 determines the default channel attributes.

Detailed Description Text (29):
The ADC device-ADC data server interface may request data from the ADC data server 130. The request produces a "data class mask" that identifies the class of data to be sent to a particular client, as discussed above. The data class masks may be device dependent or device independent. The ADC data server 130 forwards data based upon the mask(s) identified. The ADC device-ADC data server interface also provides functions for discovering specific ADC device attributes or specific client handle attributes. Device attributes include the device's enablement status. Client handle attributes include the device-dependent grid, the device-independent grid, read ahead status, and the data class read specification.

Detailed Description Text (30):
The ADC device-ADC data server interface also provides functionality for adjusting specific ADC device attributes. The ADC device-ADC data server interface includes a "query data" function that returns the number of data items stored for a client and the size of the next data item. The ADC device-ADC data server interface further performs operations such as reading a data class, setting attributes, and matching the ADC data grid. The "match grid" command requests that the ADC device handler determine if the input data matches the input grid. Both the structure and meaning of the data, and the structure and meaning of the grid are device dependent.

Detailed Description Text (31):
The ADC data server process interface provides an open function that opens a data collection device channel and returns an ADC data server client handle that allows a client to access the ADC device. The open function creates a device client handle for the ADC device. The ADC protocol handler can retrieve a device's client handle. The ADC data server process interface also allows the setting of specific ADC device attributes, such as enable/disable status; data grid (device independent and device dependent); read ahead/non-read ahead status; ADC device client handle, and data class specification.

Detailed Description Text (32):
The ADC data server process interface also provides a read function that allows an ADC protocol handler to receive ADC data or device responses from an ADC device. The read function receives as inputs an ADC data server client handle and a data class mask indicating the classes of data to be retrieved and returns as output device-dependent data. If the data class mask indicates that ADC data will be read, the read function initiates reading ADC data from the ADC device. The ADC data server 130 calls the ADC device handler's read function to notify the ADC device that a client is ready to accept data. The read function then waits until the ADC device provides the data or until a timeout arises.

Detailed Description Text (38):
The ADC data grid 303 may be stored in a memory element of the computing system 120 as a unified grid or may be stored in disparate elements throughout the computing system 120. Similarly, the computing elements that perform the tasks of the grid data matcher 302 and the data transmitter 301 may constitute unique computing elements within the ADC data server 130 or may be comprised of separate computing elements that cooperatively perform the tasks discussed. For example, the grid data

matcher 302 may be part of the ADC device-ADC data server interface discussed with regard to FIG. 2.

Detailed Description Text (53):
Local applications, such as the local application 111, may utilize the ADC SDK 633 in the collection of ADC data. Local applications may also utilize the ADC protocol handler collection 232, which may contain specialized DLLs for each ADC device configured to operate with the ADC device platform 100. The SDK 633 is a dynamic link library ("DLL") that allows ADC client applications to use legacy interfaces in communicating with ADC devices. Legacy interfaces include interfaces previously developed that provide access to one or more ADC devices. The SDK 633 allows programmers to write seamless ADC applications and then debug the applications before executing them on the ADC device platform 100.

Detailed Description Text (54):
DLLs allow executable routines to be stored separately as files having DLL extensions that are loaded only when needed by a program, such as by the local application 111. A DLL routine consumes no memory until it is used. Because a DLL routine is a separate file, a programmer may make connections or improvements to the routine without effecting the operation of the calling program or any other DLL routine. In addition, a programmer may use the same DLL routine with other programs. The DLL interface optimizes performance and resource overhead. The remote applications, such as the remote application 107, may also utilize the specialized DLLs provided by the ADC protocol handler collection 232. According to one exemplary embodiment of the invention, the ADC device platform 100 includes DLLs/COM objects for the various ADC symbologies, such as RF tag symbologies and bar code symbologies, that may be recognized by the ADC devices 117, 118. As described above, the ADC device platform 100 may be equipped with a wide variety of ADC device types.

Detailed Description Text (55):
In one exemplary embodiment, the SDK 633 supports programming elements such as Visual C/C++, Microsoft Foundation Class ("MFC"), Visual Basic, and Java. The SDK 633 may also include Active X control/Java classes, ADC device platform legacy DLLs, and the ADC data server API. Both the Active X control/Java classes and the ADC device platform legacy DLL may operate in connection with value-added reseller ("VAR") applications. For example, the VAR applications may provide data collection and unit management applications. The Active X control/Java classes may communicate with the VAR applications using Active X and Java APIs. The ADC device platform DLL communicates with the VAR applications through legacy APIs, such as a DLL interface, in one exemplary embodiment.

Detailed Description Text (58):
The SNMP master agent 629 controls the SNMP subagents in the SNMP subagent collection 624. The SNMP subagent collection 634 comprises an ADC data server SNMP subagent 627, an event log SNMP subagent 628, an ADC device SNMP subagent 625 (for the ADC device 117), and an ADC device SNMP subagent 626 (for the ADC device 118). The ADC device SNMP subagents 625, 626 respectively contain control information for the ADC devices 117, 118. The ADC device 117 and the ADC device 118 may each operate under different protocols and commands. For example, the ADC device platform 100 may be equipped with ADC devices as diverse as bar code readers and SmartCard readers. Since each ADC device typically operates under different protocols, each ADC device typically requires its own SNMP subagent. The SNMP subagent collection 634 may contain a respective ADC device SNMP subagent for each ADC device in the ADC device platform 100. Thus, the SNMP subagent collection 634 does not necessarily contain precisely two ADC device SNMP subagents. The event log SNMP subagent 628 allows the SNMP master agent 629 to retrieve the event log and set its filter. The event log SNMP subagent 628 also generates SNMP traps when specific events are received.

Detailed Description Text (60):
The SNMP architecture provides flexibility to the ADC device platform 100 by allowing the SNMP subagents to be added and removed without effecting the other SNMP subagents or the MIB collection 623. Adding a new ADC device to the ADC device platform 100 requires only adding a new SNMP subagent and storing relevant information in the MIB collection 223, regardless of the new ADC device's communication protocol. The SNMP architecture also aids ADC device platform manufacturers and their value-added resellers ("VARs") by simplifying the addition of new ADC devices. The ADC data server 130 communicates with the SNMP subagent collection 634 through a DLL interface. The ADC device platform SNMP master agent 629 also communicates with the ADC data server, the network communication unit 221, and the computing system's operating system.

Detailed Description Text (66):
Network Driver Interface Specification ("NDIS") Version 4.0 723, provides hardware and protocol independence for network drivers utilized by the ADC device platform 100. NDIS, of which version 4.0 may be used, offers a device driver standard that allows for running multiple protocols on the same network adapter.

Detailed Description Text (67):
Legacy radio driver 724 refers to pre-existing radio driver protocols that may be utilized within the ADC device platform 100. Open Radio Interface 725 provides radio driver interfaces that are customizable across radio devices. An Ethernet driver 726 enables Ethernet communications for the ADC device platform 100. The Ethernet provides a local area network ("LAN") that connects computing elements together within the same building or campus. The Ethernet is a physical link and data link protocol, reflecting the two lowest layers of the OSI model. Point-to-Point Protocol ("PPP") 727 is a data link protocol that provides a well-known method for transmitting IP frames over a circuit. The PPP 727 may communicate with a WindowsCE built-in serial port driver 728 that further processes communications into the physical layer 708.

Detailed Description Text (69):
Legacy radio interface 730 provides a match at the physical layer 708 for the legacy radio driver 724. Similarly, Open Radio Hardware Interfaces 731 provides a match at the physical layer 708 for the Open Radio Interface Protocol Driver 725. Ethernet controller 732 matches with the Ethernet Driver 726, and Serial Port (COM1) 733 matches with the WindowsCE built-in serial driver 728.

Detailed Description Text (76):
According to one embodiment of the invention, the data communications network may use Java applets as the user interface to communicate data requests, including directions to the ADC data grid, to ADC device platforms. Java is an object-oriented programming language similar to C++. Java was designed to be secure and platform neutral, meaning that Java code may run on any computing platform. Java is a useful language for programming applications for the World Wide Web since users access the web from many different types of computers. Java is especially well adapted for use in programming small applications, or applets, for use in the World Wide Web. A Java applet may be loaded and run by an already running Java application, such as a web browser. Java applets may be downloaded and run by any web browser capable of interpreting Java, such as Microsoft Internet Explorer, Netscape Navigator, and Hot Java.

CLAIMS:

7. The method of claim 1 wherein the at least one ADC device is one of a bar code reader, a radio frequency ("RF") tag reader, a resonator reader, a SmartCard reader, a magnetic stripe reader, a two-dimensional symbol reader, an optical character recognition ("OCR") reader, a dipole device reader, and a speech input recognizing device.

19. The system of claim 16 wherein a data type for the data set of the plurality of data sets comprises one of bar code data, radio frequency ("RF") tag data, resonator data, SmartCard data, magnetic stripe data, optical character recognition ("OCR") data, text data, two-dimensional symbol data, dipole device data, and speech input data.

20. The system of claim 16 wherein at least one ADC device of the plurality of ADC devices is one of a bar code reader, a radio frequency ("RF") tag reader, a resonator reader, a SmartCard reader, a magnetic stripe reader, an optical character recognition ("OCR") reader, a two-dimensional symbol reader, a dipole device reader, and a speech input recognizing device.

30. The system of claim 28 wherein a data type for the data set of the plurality of data sets comprises one of bar code data, radio frequency ("RF") tag data, resonator data, SmartCard data, magnetic stripe data, optical character recognition ("OCR") data, two-dimensional symbol data, text data, dipole device data, and speech input data.

31. The system of claim 28 wherein at least one ADC device of the plurality of ADC devices is one of a bar code reader, a radio frequency ("RF") tag reader, a resonator reader, a SmartCard reader, a magnetic stripe reader, an optical character recognition ("OCR") reader, a dipole device reader, two-dimensional symbol reader and a speech input recognizing device.

L19: Entry 9 of 13 File: USPT May 7, 2002

DOCUMENT-IDENTIFIER: US 6385309 B1
** See image for Certificate of Correction **
TITLE: System and method for storing and transferring information tokens in a communication network

Detailed Description Text (8):
The CPEs 12, 24 can communicate with the VOD modems 14, 22, respectively, using a standard interface, such as an RS-232 interface, a personal computer (PC) parallel port, a PC bus, a universal serial bus, or the like. The VOD modems 14, 22 can communicate with the COs 16, 20 using a conventional analog local-loop, an integrated services digital network (ISDN) interface, or the like. The modems 14, 22 can include conventional jacks or connectors for providing a detachable interface to the telephone network 15.

Detailed Description Text (9):
The VOD modems 14, 22 can be implemented using commercially-available VOD modems, such as the MRi-1456 advanced simultaneous voice-over-data (ASVD) modem, available from MRi (UK) Ltd., of Wembley, England. Such a modem can be connected to the CPEs 12 or 24 using a conventional peripheral components interface (PCI) bus.

Detailed Description Text (19):
FIG. 3 shows a detailed block diagram of the caller CPE 12 of FIG. 1. The CPE 12 can include a microprocessor (uP) 60, a memory 64, a modem interface 62, a smartcard interface 64 and a bus 67. Also included in the CPE 12 is a telephony circuit 68 for providing conventional analog phone service.

Detailed Description Text (20):
The microprocessor 60 can be any microprocessor, microcontroller, or the like, such as one from the .times.86 family of microprocessors from Intel, Corp., or the PowerPC.TM. family of microprocessors from Motorola, Inc. The bus 67 can be a conventional microprocessor bus such as a peripheral component interface (PCI) bus, ISA bus, ESA bus, or the like. The memory 64 can be any type of computer memory, such as a random access memory (RAM), flash memory, hard drive, zip drive, floppy drive, or the like.

Detailed Description Text (21):
The modem interface 62 permits the CPE 12 to communicate with the VOD modem 14. Although the modem interface 62 is shown as being included in the CPE 12, one of ordinary skill in the art will readily understand that the modem interface 62 can be included in the VOD modem 14 itself, or alternatively, that the VOD modem 14 can be included internally within the CPE 12.

Detailed Description Text (22):
Cookie files can be stored in the memory 64, or alternatively, in an external memory, such as a smartcard 66. The smartcard 66 can be any commercially-available smartcard, contactless or contact, insertable into smartcard interface 65 of the CPE 12, such as a Multi-Function Card MCF/4K, from IBM Corporation. The smartcard interface 65 can include a commercially available smartcard reader, such as the GCI400 Smartcard Reader, from Gemplus, Corp. for reading ISO 7816 compliant smartcards. One of ordinary skill in the art will appreciate that the GCI400 can be

readily configured to interface to a conventional microprocessor bus, such as the bus 67.

Detailed Description Text (23):
The telephony circuit 68 can include conventional circuitry for providing analog telephone service. Voice signals received from the local-loop interface are converted by the telephony circuit 68 for audible presentation to the caller. In addition, the circuit 68 can provide standard end device functions, such as ring detection and generation, dual-tone multi-frequency (DTMF) dialing, line termination, power supply conditioning, and the like. The telephony circuit 68 can include an ARCOFI Chip, Part No. PSB2163, manufactured by Siemens Corporation. In such an embodiment, the ARCOFI Chip can be readily interfaced to the microprocessor 60. The ARCOFI Chip also provides an interface to a standard loudspeaker 69.

Detailed Description Text (26):
In step 71, the CPE 12 monitors the modem interface 62 for a cookie read-request transmitted by the called party CPE 24. The read-request can include an identifier corresponding to a particular cookie file or template stored in CPE 12. Upon receiving the read-request, the microprocessor 60 can retrieve the requested cookie file from either the smartcard 66 or memory 64 (step 72). The cookie file is then transferred via the telephone network 15 to the called CPE 24.

Detailed Description Text (27):
In step 73, the CPE 12 monitors the modem interface for a cookie file write-request. A write-request is transmitted by the called party CPE 24 indicating that it is ready to transmit a modified cookie file for storage in the caller CPE 12. In step 74, the CPE 12 receives and stores the modified cookie file. The cookie file is received by the CPE 12 at the modem interface 62. The microprocessor 60 causes the modem interface 62 to transfer the incoming cookie file via the bus 67 to either the smartcard storage 66 or the memory 64.

Detailed Description Text (33):
Alternately, a symmetrical DSL modem can be used in place of the ADSL modem. As one of ordinary skill in the art will readily appreciate, in such an arrangement, splitters are typically included in the CPEs and COs to filter data signals and voice.

Detailed Description Text (36):
The AIN 202 includes a service management system (SMS) 204, a switching control point (SCP) 206, a signal transfer point (STP) 208, a switching service point (SSP) 210, and a service node/intelligent peripheral (SN/IP) 212. The AIN uses common channel signalling (CCS) for communication between the SMS, SCP, STP, and SSP. CCS is an out-of-band signalling method that utilizes packet-switched networking to allow messages to be transported on a dedicated high-speed data network, separate from the subscriber voice and data communications path. The CCS utilizes the Signalling System No. 7 (SS7) protocol to send messages between the AIN elements regarding call setup, line status, caller identification, and other network services, including AIN inquiries. The use of SS7 in an AIN is well known to those skilled in the art. Also, as is known in the art, the SNIIP 212 and SSP 210 can communicate using an integrated services digital network (ISDN) interface.

Detailed Description Text (41):
In an alternative arrangement, the CPE 24 can communicate with the switch 214 using an ISDN interface. In this arrangement, voice data can be carried on the bearer channel of the ISDN connection, while cookie retrieval signalling can be carried on the signalling channel of the ISDN. This permits simultaneous voice and data transfer to/from the CPE 24. The CPE 24 can be adapted to transmit SS7 messages via the ISDN interface. These messages can be directly passed from the switch 214 to the STP 208 over SS7 link 209. The SS7 messages can contain caller information and information about CPE 24 for retrieving a cookie file from the SCP 206 specific to

the CPE 24 and the particular caller. Upon receiving the message, the STP 208 forwards the request to the SCP 206. In response, the SCP 206 returns a cookie file to the CPE 24 using the SS7 channel.

CLAIMS:

12. The telecommunications network as recited in claim 11, wherein said AIN includes an integrated services digital network (ISDN) interface.

First Hit Fwd Refs

Generate Collection Print

L19: Entry 2 of 13 File: USPT Nov 18, 2003

DOCUMENT-IDENTIFIER: US 6650710 B1
TITLE: Packet filtering

Abstract Text (1):
A method of filtering a packet data stream as transmitted in a digital audiovisual transmission system characterized in that transport packet data is filtered at a first stage by a digital filter according to the characteristics of the transport packet header, selected filtered data being then passed directly to a memory element within the receiver/decoder. In a second aspect, the method comprises filtering a packet data stream characterized in that transport packet data is filtered at a first stage by a first digital filter according to the characteristics of the transport packet header, selected filtered data from the paload corresponding to a continuous flux of data then being passed to a second digital filter for a second stage of filtering.

Brief Summary Text (13):
According to one aspect of the present invention, there is provided a method of filtering a packet data stream as transmitted in an digital audiovisual transmission system and received by a receiver/decoder characterised in that transport packet data is filtered at a first stage by a first digital filter according to the characteristics of the transport packet header, selected filtered data then being passed directly to a memory element within the receiver/decoder.

Brief Summary Text (14):
This method of treating the packet data represents a radical departure to the conventional method, in which data filtered according to the PID packet header value is either sent directly (in the form of a PES stream) to a dedicated processor or is filtered in a second stage according to a TID or PSI header value before being assembled in the memory of the decoder.

Brief Summary Text (23):
Data filtered by the second digital filter may be passed to an associated memory element for storage and/or to a processor for decoding after having been filtered.

Brief Summary Text (24):
According to a second aspect, the present invention comprises a method of filtering a packet data stream as transmitted in an digital audiovisual transmission system and received by a receiver/decoder characterised in that transport packet data is filtered at a first stage by a first digital filter according to the characteristics of the transport packet header, selected filtered data from the payload corresponding to a continuous flux of data then being passed to a second digital filter for a second stage of filtering.

Brief Summary Text (28):
As before, data filtered by the second digital filter may be passed to an associated memory element for storage and/or to a processor for execution, depending on the results of the filtering.

Brief Summary Text (31):
The present invention extends in one aspect to a receiver/decoder for a digital

audiovisual transmission system comprising a first digital filter and memory element, characterised in that selected transport packet data is filtered at a first stage by the first digital filter according to the characteristics of the transport packet header and thereafter passed directly to the memory element.

Brief Summary Text (32):
The present invention equally extends to a receiver/decoder for a digital audiovisual transmission system comprising a first and second digital filter, characterised in that transport packet data is filtered at a first stage by the first digital filter, selected filtered data from the payload corresponding to a continuous flux of data then being passed to the second digital filter for a second stage of filtering.

Brief Summary Text (33):
In the context of the present application the term "digital audiovisual transmission system" refers to all transmission systems for transmitting or broadcasting primarily audiovisual or multimedia digital data. Whilst the present invention is particularly applicable to a broadcast digital television system such as a satellite, terrestrial or cable television system, the present invention may equally be used in filtering data sent by a fixed telecommunications network for multimedia internet applications, combination telecommunication and broadcast systems etc.

Detailed Description Text (5):
A conditional access system 15 is connected to the multiplexer 4 and the receiver/decoder 13, and is located partly in the broadcast centre and partly in the decoder. It enables the end user to access digital television broadcasts from one or more broadcast suppliers. A smartcard, capable of deciphering messages relating to commercial offers (that is, one or several television programmes sold by the broadcast supplier), can be inserted into the receiver/decoder 13. Using the decoder 13 and smartcard, the end user may purchase commercial offers in either a subscription mode or a pay-per-view mode.

Detailed Description Text (10):
The decoder 13 comprises a central processor 20 including associated memory elements and adapted to receive input data from a serial interface 21, a parallel interface 22, a modem 23 (connected to the modem back channel 17 of FIG. 1), and switch contacts 24 on the front panel of the decoder.

Detailed Description Text (11):
The decoder is additionally adapted to receive inputs from an infra-red remote control via a control unit 26 and also possesses two smartcard readers 27, 28 adapted to read bank or subscription smartcards 29, 30 respectively. The subscription smartcard reader 28 engages with an inserted subscription card 30 and with a conditional access unit 29 to supply the necessary control word to a demultiplexer/descrambler 30 to enable the encrypted broadcast signal to be descrambled. The decoder also includes a conventional tuner 31 and demodulator 32 to receive and demodulate the satellite transmission before being filtered and demultiplexed by the unit 30.

Detailed Description Text (12):
Processing of data within the decoder is generally handled by the central processor 20. The software architecture of the central processor may correspond to that used in a known decoder and will not be described here in any detail. It may be based, for example, on a virtual machine interacting via an interface layer with a lower level operating system implemented in the hardware components of the decoder. In terms of the hardware architecture, the decoder will be equipped with a processor, memory elements such as ROM, RAM, FLASH etc. as in known decoders.

Detailed Description Text (23):

By way of contrast, the filters 42 according to this embodiment of the invention carry out a filtering of data based on the whole of the packet header 63, 64 including the PID section 63 but excluding the synchronisation byte 62. For this reason, the filters 42 each have a size of 24 bits or 3 bytes.

Detailed Description Text (34):
Referring to FIG. 3, in order to correctly filter data based on the PES Header values described above, the second filter 45 needs to have a size of 82 bits corresponding to the maximum size of all the above values. The filtering process can be carried out on an "OR" basis, i.e. if any of the fields (flag, PTS, DTS, DSM Trick Mode) are present the PES stream may be sent to a memory element 46 or via a switch element 47 to the dedicated processor element for real time processing and presentation. As before, filtering may be carried out according to known digital filtering techniques; masking, comparison of a field to a given maximum/minimum value etc.

Detailed Description Text (39):
As mentioned in the introduction, this filtering of data by table ID is in itself known and described, for example, in the pending patent application PCI/EP97/02114 filed in the name of the present applicants. As such, it need not be described in any further detail.

L19: Entry 3 of 13 File: USPT Nov 4, 2003

DOCUMENT-IDENTIFIER: US 6643361 B2
TITLE: System and method for storing and transferring information tokens in a communication network

Detailed Description Text (8):
The CPEs 12, 24 can communicate with the VOD modems 14, 22, respectively, using a standard interface, such as an RS-232 interface, a personal computer (PC) parallel port, a PC bus, a universal serial bus, or the like. The VOD modems 14, 22 can communicate with the COs 16, 20 using a conventional analog local-loop, an integrated services digital network (ISDN) interface, or the like. The modems 14, 22 can include conventional jacks or connectors for providing a detachable interface to the telephone network 15.

Detailed Description Text (9):
The VOD modems 14, 22 can be implemented using commercially-available VOD modems, such as the MRi-1456 advanced simultaneous voice-over-data (ASVD) modem, available from MRi (UK) Ltd., of Wembley, England. Such a modem can be connected to the CPEs 12 or 24 using a conventional peripheral components interface (PCI) bus.

Detailed Description Text (19):
FIG. 3 shows a detailed block diagram of the caller CPE 12 of FIG. 1. The CPE 12 can include a microprocessor (uP) 60, a memory 64, a modem interface 62, a smartcard interface 64 and a bus 67. Also included in the CPE 12 is a telephony circuit 68 for providing conventional analog phone service.

Detailed Description Text (20):
The microprocessor 60 can be any microprocessor, microcontroller, or the like, such as one from the x86 family of microprocessors from Intel, Corp., or the PowerPC.TM. family of microprocessors from Motorola, Inc. The bus 67 can be a conventional microprocessor bus such as a peripheral component interface (PCI) bus, ISA bus, ESA bus, or the like. The memory 64 can be any type of computer memory, such as a random access memory (RAM), flash memory, hard drive, zip drive, floppy drive, or the like.

Detailed Description Text (21):
The modem interface 62 permits the CPE 12 to communicate with the VOD modem 14. Although the modem interface 62 is shown as being included in the CPE 12, one of ordinary skill in the art will readily understand that the modem interface 62 can be included in the VOD modem 14 itself, or alternatively, that the VOD modem 14 can be included internally within the CPE 12.

Detailed Description Text (22):
Cookie files can be stored in the memory 64, or alternatively, in an external memory, such as a smartcard 66. The smartcard 66 can be any commercially-available smartcard, contactless or contact, insertable into smartcard interface 65 of the CPE 12, such as a Multi-Function Card MCF/4K, from IBM Corporation. The smartcard interface 65 can include a commercially available smartcard reader, such as the GC1400 Smartcard Reader, from Gemplus, Corp. for reading ISO 7816 compliant smartcards. One of ordinary skill in the art will appreciate that the GC1400 can be readily configured to interface to a conventional microprocessor bus, such as the

bus 67.

Detailed Description Text (23):
The telephony circuit 68 can include conventional circuitry for providing analog telephone service. Voice signals received from the local-loop interface are converted by the telephony circuit 68 for audible presentation to the caller. In addition, the circuit 68 can provide standard end device functions, such as ring detection and generation, dual-tone multi-frequency (DTMF) dialing, line termination, power supply conditioning, and the like. The telephony circuit 68 can include an ARCOFI Chip, Part No. PS82163, manufactured by Siemens Corporation. In such an embodiment, the ARCOFI Chip can be readily interfaced to the microprocessor 60. The ARCOFI Chip also provides an interface to a standard loudspeaker 69.

Detailed Description Text (26):
In step 71, the CPE 12 monitors the modem interface 62 for a cookie read-request transmitted by the called party CPE 24. The read-request can include an identifier corresponding to a particular cookie file or template stored in CPE 12. Upon receiving the read-request, the microprocessor 60 can retrieve the requested cookie file from either the smartcard 66 or memory 64 (step 72). The cookie file is then transferred via the telephone network 15 to the called CPE 24.

Detailed Description Text (27):
In step 73, the CPE 12 monitors the modem interface for a cookie file write-request. A write-request is transmitted by the called party CPE 24 indicating that it is ready to transmit a modified cookie file for storage in the caller CPE 12. In step 74, the CPE 12 receives and stores the modified cookie file. The cookie file is received by the CPE 12 at the modem interface 62. The microprocessor 60 causes the modem interface 62 to transfer the incoming cookie file via the bus 67 to either the smartcard storage 66 or the memory 64.

Detailed Description Text (33):
Alternately, a symmetrical DSL modem can be used in place of the ADSL modem. As one of ordinary skill in the art will readily appreciate, in such an arrangement, splitters are typically included in the CPEs and COs to filter data signals and voice.

Detailed Description Text (36):
The AIN 202 includes a service management system (SMS) 204, a switching control point (SCP) 206, a signal transfer point (STP) 208, a switching service point (SSP) 210, and a service node/intelligent peripheral (SN/IP) 212. The AIN uses common channel signalling (CCS) for communication between the SMS, SCP, STP, and SSP. CCS is an out-of-band signalling method that utilizes packet-switched networking to allow messages to be transported on a dedicated high-speed data network, separate from the subscriber voice and data communications path. The CCS utilizes the Signalling System No. 7 (SS7) protocol to send messages between the AIN elements regarding call setup, line status, caller identification, and other network services, including AIN inquiries. The use of SS7 in an AIN is well known to those skilled in the art. Also, as is known in the art, the SN/IP 212 and SSP 210 can communicate using an integrated services digital network (ISDN) interface.

Detailed Description Text (41):
In an alternative arrangement, the CPE 24 can communicate with the switch 214 using an ISDN interface. In this arrangement, voice data can be carried on the bearer channel of the ISDN connection, while cookie retrieval signalling can be carried on the signalling channel of the ISDN. This permits simultaneous voice and data transfer to/from the CPE 24. The CPE 24 can be adapted to transmit SS7 messages via the ISDN interface. These messages can be directly passed from the switch 214 to the STP 208 over SS7 link 209. The SS7 messages can contain caller information and information about CPE 24 for retrieving a cookie file from the SCP 206 specific to the CPE 24 and the particular caller. Upon receiving the message, the STP 208

forwards the request to the SCP 206. In response, the SCP 206 returns a cookie file to the CPE 24 using the SS7 channel.

CLAIMS:

9. The method of claim 1, wherein the first CPE comprises a smart card interface for storing the modified data file on a smart card.

20. The method of claim 11, wherein storing the electronically modified data file comprises storing the electronically modified data file on a smart card coupled to a smart card interface of the first CPE.

First Hit Fwd Refs

Generate Collection Print

L19: Entry 12 of 13 File: USPT Nov 28, 2000

DOCUMENT-IDENTIFIER: US 6154528 A
TITLE: System and method for storing and transferring information tokens in a low network communication

Detailed Description Text (8):
The CPEs 12, 24 can communicate with the VOD modems 14, 22, respectively, using a standard interface, such as an RS-232 interface, a personal computer (PC) parallel port, a PC bus, a universal serial bus, or the like. The VOD modems 14, 22 can communicate with the COs 16, 20 using a conventional analog local-loop, an integrated services digital network (ISDN) interface, or the like. The modems 14, 22 can include conventional jacks or connectors for providing a detachable interface to the telephone network 15.

Detailed Description Text (9):
The VOD modems 14, 22 can be implemented using commercially-available VOD modems, such as the MRi-1456 advanced simultaneous voice-over-data (ASVD) modem, available from MRi (UK) Ltd., of Wembley, England. Such a modem can be connected to the CPEs 12 or 24 using a conventional peripheral components interface (PCI) bus.

Detailed Description Text (19):
FIG. 3 shows a detailed block diagram of the caller CPE 12 of FIG. 1. The CPE 12 can include a microprocessor (uP) 60, a memory 64, a modem interface 62, a smartcard interface 64 and a bus 67. Also included in the CPE 12 is a telephony circuit 68 for providing conventional analog phone service.

Detailed Description Text (20):
The microprocessor 60 can be any microprocessor, microcontroller, or the like, such as one from the x86 family of microprocessors from Intel, Corp., or the PowerPC.TM. family of microprocessors from Motorola, Inc. The bus 67 can be a conventional microprocessor bus such as a peripheral component interface (PCI) bus, ISA bus, ESA bus, or the like. The memory 64 can be any type of computer memory, such as a random access memory (RAM), flash memory, hard drive, zip drive, floppy drive, or the like.

Detailed Description Text (21):
The modem interface 62 permits the CPE 12 to communicate with the VOD modem 14. Although the modem interface 62 is shown as being included in the CPE 12, one of ordinary skill in the art will readily understand that the modem interface 62 can be included in the VOD modem 14 itself, or alternatively, that the VOD modem 14 can be included internally within the CPE 12.

Detailed Description Text (22):
Cookie files can be stored in the memory 64, or alternatively, in an external memory, such as a smartcard 66. The smartcard 66 can be any commercially-available smartcard, contactless or contact, insertable into smartcard interface 65 of the CPE 12, such as a Multi-Function Card MCF/4K, from IBM Corporation. The smartcard interface 65 can include a commercially available smartcard reader, such as the GCI400 Smartcard Reader, from Gemplus, Corp. for reading ISO 7816 compliant smartcards. One of ordinary skill in the art will appreciate that the GCI400 can be readily configured to interface to a conventional microprocessor bus, such as the

bus 67.

Detailed Description Text (23):
The telephony circuit 68 can include conventional circuitry for providing analog telephone service. Voice signals received from the local-loop interface are converted by the telephony circuit 68 for audible presentation to the caller. In addition, the circuit 68 can provide standard end device functions, such as ring detection and generation, dual-tone multi-frequency (DTMF) dialing, line termination, power supply conditioning, and the like. The telephony circuit 68 can include an ARCOFI Chip, Part No. PSB2163, manufactured by Siemens Corporation. In such an embodiment, the ARCOFI Chip can be readily interfaced to the microprocessor 60. The ARCOFI Chip also provides an interface to a standard loudspeaker 69.

Detailed Description Text (26):
In step 71, the CPE 12 monitors the modem interface 62 for a cookie read-request transmitted by the called party CPE 24. The read-request can include an identifier corresponding to a particular cookie file or template stored in CPE 12. Upon receiving the read-request, the microprocessor 60 can retrieve the requested cookie file from either the smartcard 66 or memory 64 (step 72). The cookie file is then transferred via the telephone network 15 to the called CPE 24.

Detailed Description Text (27):
In step 73, the CPE 12 monitors the modem interface for a cookie file write-request. A write-request is transmitted by the called party CPE 24 indicating that it is ready to transmit a modified cookie file for storage in the caller CPE 12. In step 74, the CPE 12 receives and stores the modified cookie file. The cookie file is received by the CPE 12 at the modem interface 62. The microprocessor 60 causes the modem interface 62 to transfer the incoming cookie file via the bus 67 to either the smartcard storage 66 or the memory 64.

Detailed Description Text (33):
Alternately, a symmetrical DSL modem can be used in place of the ADSL modem. As one of ordinary skill in the art will readily appreciate, in such an arrangement, splitters are typically included in the CPEs and COs to filter data signals and voice.

Detailed Description Text (36):
The AIN 202 includes a service management system (SMS) 204, a switching control point (SCP) 206, a signal transfer point (STP) 208, a switching service point (SSP) 210, and a service node/intelligent peripheral (SN/IP) 212. The AIN uses common channel signalling (CCS) for communication between the SMS, SCP, STP, and SSP. CCS is an out-of-band signalling method that utilizes packet-switched networking to allow messages to be transported on a dedicated high-speed data network, separate from the subscriber voice and data communications path. The CCS utilizes the Signalling System No. 7 (SS7) protocol to send messages between the AIN elements regarding call setup, line status, caller identification, and other network services, including AIN inquiries. The use of SS7 in an AIN is well known to those skilled in the art. Also, as is known in the art, the SN/IP 212 and SSP 210 can communicate using an integrated services digital network (ISDN) interface.

Detailed Description Text (41):
In an alternative arrangement, the CPE 24 can communicate with the switch 214 using an ISDN interface. In this arrangement, voice data can be carried on the bearer channel of the ISDN connection, while cookie retrieval signalling can be carried on the signalling channel of the ISDN. This permits simultaneous voice and data transfer to/from the CPE 24. The CPE 24 can be adapted to transmit SS7 messages via the ISDN interface. These messages can be directly passed from the switch 214 to the STP 208 over SS7 link 209. The SS7 messages can contain caller information and information about CPE 24 for retrieving a cookie file from the SCP 206 specific to the CPE 24 and the particular caller. Upon receiving the message, the STP 208

forwards the request to the SCP 206. In response, the SCP 206 returns a cookie file to the CPE 24 using the SS7 channel.

CLAIMS:

13. The intelligent telephone of claim 10, further including a smartcard reader capable of storing the modified data file on a smartcard.

14. An intelligent telephone, comprising:

a bus;

a memory, operatively coupled to the bus, for storing at least one data file including caller profile information;

a microprocessor operatively coupled to the bus;

an interface capable of being operatively coupled to a telecommunications network;

a modem, operatively coupled to the interface and the bus, said modem for transferring the at least one data file to the telecommunications network via the interface simultaneously with voice data;

means for responding to an electronic request for a data file;

a telephony circuit, in communication with the interface, for generating signals representing voice; and

a software program routine, executable by the microprocessor, for causing the requested data file to be transferred from the memory to the modem in response to said electronic request received from a called party.

16. The intelligent telephone of claim 14, further including a smartcard interface in communication with the bus capable of storing the at least one data file on a smartcard.

L5: Entry 3 of 4 File: USPT Sep 30, 2003

DOCUMENT-IDENTIFIER: US 6629081 B1
** See image for Certificate of Correction **
TITLE: Account settlement and financing in an e-commerce environment

Detailed Description Text (2182):
This technique filters the information or content displayed to the user based on what is entered by, or known about the user. Many personalized sites use some form of content filtering. A variety of different filtering techniques have emerged. Some are adaptations of traditional client/server techniques adapted to the Web. In the first grouping of techniques presented here, the user controls how the information is filtered. User-controlled, explicit content filtering. Techniques in this category enable the user to filter data or content. Personalized Information. Allows the user to see information specific to them. The content in this case may be information about the user's profile, about products they have or a past order history. This technique allows the user to filter the data by building `where clause` statement or execute `canned` queries. It is often used when the user is familiar with the data and wishes to organize or filter it in multiple ways. User-controlled, implicit content filtering. Collaborative filtering. Collaborative filtering determines clusters of users with similar interests, either by asking users explicitly or by observing user selections and actions to determine those interests implicitly. This is an effective technique for creating recommendations for products. Site controlled content filtering. Contextual Inferences. Contextual inference uses human-determined rules to select content based on behavior, interest or other profile attributes. What's New. Based on knowing when the user last visited, the site determines what content has changed since the last time the user has interacted with the site and display a list of changes. A personalized "what's new" is far more effective than a generic "what's new" that is shown to all users--obviously what's new is different for the user who was here yesterday versus the one who hasn't used the site in six months. The content that is marked as new may be further constrained to only contain the areas that the user has interacted with in the past.

Detailed Description Text (2193):
Push technologies enable an enterprise to reach and provide value to their customer outside of the traditional interactions. Of the personalization techniques described above, content filtering doesn't require that it be delivered though an interactive session. Keep in mind that push does not necessarily mean personalized. A site may simple push the same information to all users or subscribers. Personalized push refers to information that is filtered based on a specific user's request or profile or where the delivery is scheduled for a specific user. There are a variety of different legitimate business reasons to push a users information. The following are some examples of personalized push: Information Delivery. A user subscribes to receive information on a scheduled basis. The information that is pushed is either determined through user selection--the user selects the subjects and type of information that they wish to receive, or the site determines what information the user may be interested in receiving based on their profile information. For example, at an investment web site, users are allowed to subscribe to investment information feeds. The user decided to receive a daily feed of the stock quotes for the commodities they hold. The site pushes this information and additionally pushes news articles and stock recommendations based on the user's

portfolio holdings. Event Reminders. The user subscribes to receive event reminders from the web site. An event reminder might remind the user of specific dates of interest such as a birthday or anniversary, a bill reminder or that an action is required in the near future. Information Update. A site may also push updates to a user. For example, Microsoft's Expedia allows users to request fare updates. A user can choose a specific destination of interest. If the fare to this location changes, the user is notified of the fare update.

Detailed Description Text (2336):
Payment Methods There are a multitude of different vendors and technologies available for handling electronic payments. The infrastructure, process, and technology may vary dramatically from vendor to vendor. The actual mediums for the current payment options fall into these categories: Credit-Based Payment. Today, the most widely-used electronic payment option is the credit card. With the new transaction protocols and security features, credit cards can be used on the Internet just as they are in the real world. Consumer confidence is higher with the already familiar standard. Current overhead for clearing, settlement and fraud makes credit card based solutions uneconomical for transactions of small dollar amounts. Debit-Based Payment. Payment utilizing this method will directly debit and credit accounts. These may take the form of debit cards, electronic checks or messages utilizing EDI or EFT. Electronic Cash. Electronic cash is the electronic equivalent of real paper cash. It is usually implemented using public-key cryptography, digital signatures and blind signatures. Electronic cash is "digital" money on the computer's hard disk. Theoretically, the money could be spent in very small increments, such as tenths of a cent (U.S.) or less. In an electronic cash system there is usually a bank, responsible for issuing currency, consumers that obtain cash from either banks or brokers and merchants who will accept the digital cash for goods and services. In short, the bank, merchant and consumer each own a public and private key which is used to encrypt and digitally sign the electronic cash. Smartcards. A smartcard is a programmable storage device the same in size and appearance as a normal credit card. It contains a microchip to store and process information. Some of these cards can contain stored value in the form of digital coins. A lost card means lost value, just like cash. The person holding the card can spend the value stored on it at any merchant accepting smartcards. This technology is particularly useful for online shopping, and is far less vulnerable than systems storing value on a hard disk. Transaction costs for this form of payment are very low, enabling the user to conduct micro-transactions of one penny or less. Microsoft and several computer manufacturers are pushing for standards to incorporate smartcard readers into PC keyboards, and most TV set-top Internet access devices already have them Digital Wallets. Digital wallet software facilitates secure, online transactions between the consumer and the merchant, and between the merchant and the bank. For the consumer, there will soon be literally hundreds of software "digital wallets" available. They will likely be free and similar in function, running within a web browser. Payment Authorization. In many cases, consumer sites which implement a credit card payment method will require payment authorization. In some cases the actual settlement process can not occur until the items are shipped. JEPI, being developed by W3C and CommerceNet in cooperation with many large technology companies, is a standard mechanism for Web clients and servers to find out what payment capabilities they have in common, and negotiate the payment instrument, protocol, and transport between one another. This will be transparent to the user; they will simply be told by the wallet software what payment options are available at this merchant (along with any available discounts for payment type or membership affiliations), and asked to choose.